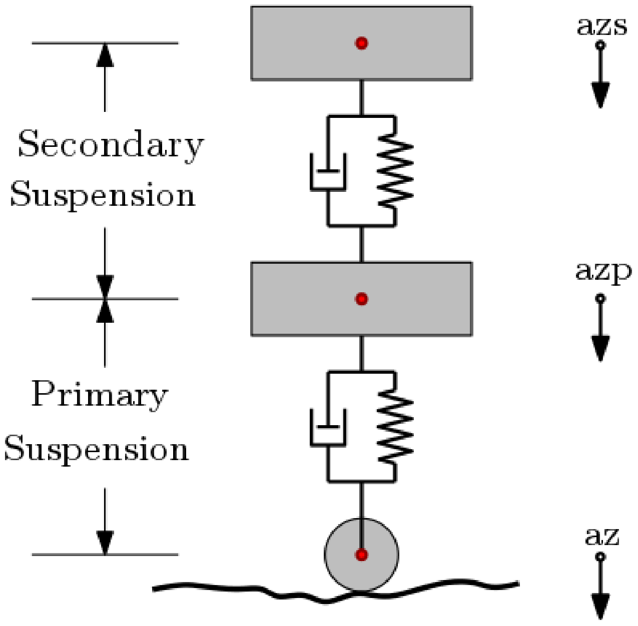
Domain Knowledge
Vehicle Dynamics Bogies

Bernhard Girstmair

SIEMENS

Acceleration level



MEMS Sensors (can measure DC)
Signal peaks up to 10-20 m/s²

MEMS Sensors (can measure DC)
Signal peaks up to 100 m/s²

ICP Sensors (can not measure DC)
Signal peaks up to 1000 m/s²

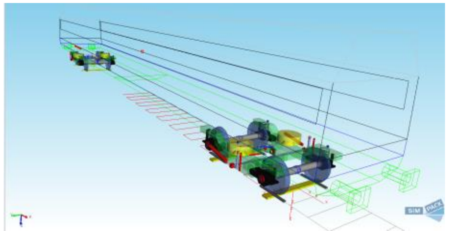
- Suspension elements act like low pass filters.
- Compared to a car there is a primary suspension needed because of the rigid wheel/rail contact, instead of a soft tyre.

Simplification of a Railway Vehicle

Real Vehicle

Real world application
Dependency on operational conditions
Variance of nominal values of components

Non-linear 3D multibody-simulation
>50 degrees of freedom (DOF)



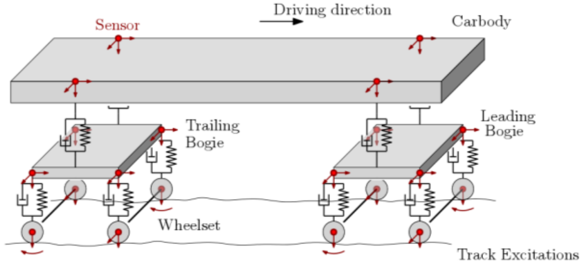
SW Tool Simpack

State of the art for engineering procedures

Simulation takes long time

Uncomfortable Interface of Simpack to Python/Matlab

Linear 3D multibody-simulation
4*2+2*5+5=23 DOF



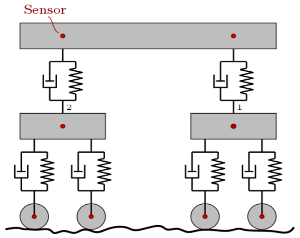
Differential equations solved by Odeint using Python/Matlab

Already good approximation of vehicle dynamics

Simulation time is acceptable

Can be implemented in Python/Matlab

Linear 2D multibody-simulation
6 DOF



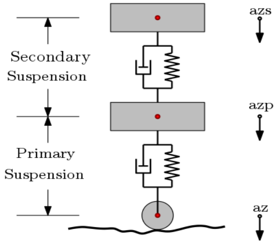
Differential equations solved by Odeint using Python/Matlab

Can capture main effects (incl kinematics) of vehicle dynamics

Simulation time is good

Can be implemented in Python/Matlab

Linear 1D multibody-simulation
2 DOF



Differential equations solved by Odeint using Python/Matlab

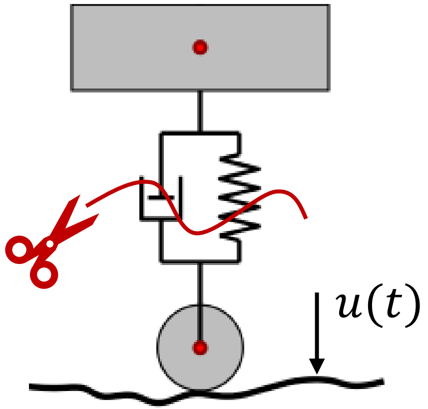
Can capture main effects of vehicle dynamics

Simulation time is fast

Can be implemented in Python/Matlab

SIEMENS

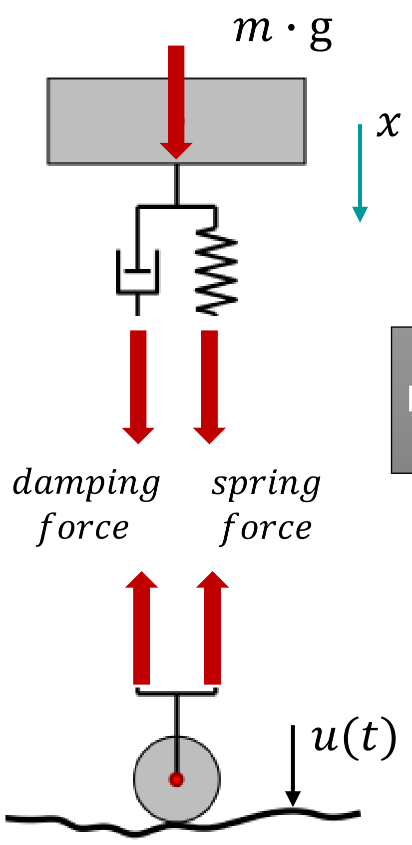
1-mass oscillator



$$\dot{u}(t) = \int \ddot{x}_{Axlebox}\, dt$$

$$u(t) = \int\int \ddot{x}_{Axlebox}\, dt\, dt$$

Free body diagram



Equations

$$spring - force: \quad FC = c \cdot \Delta x \quad (re - centering)$$

$$damping - force: \quad FD = d \cdot \Delta \dot{x} \quad (re - centering)$$

$$Newtons - laws: \quad m \cdot \ddot{x} = \sum F, \qquad J \cdot \ddot{\varphi} = \sum M$$

$$actio = reactio$$

$$m \cdot \ddot{x} = m \cdot g + F_D + F_C$$

$$m \cdot \ddot{x} = \cancel{m \cdot g} + d \cdot (\dot{u} - \dot{x}) + c \cdot (u - x)$$

*) Steady state

$$m \cdot \ddot{x} + d \cdot \dot{x} + c \cdot x = d \cdot \dot{u} + c \cdot u$$

$$Standard\ solver\ require: \dot{\boldsymbol{y}} = F(\boldsymbol{y}, t)$$

$$State - space\ transformation: \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x \\ \dot{x} \end{pmatrix} \qquad \begin{aligned} \dot{y}_1 &= \dot{x} = y_2 \\ \dot{y}_2 &= \ddot{x} \end{aligned}$$

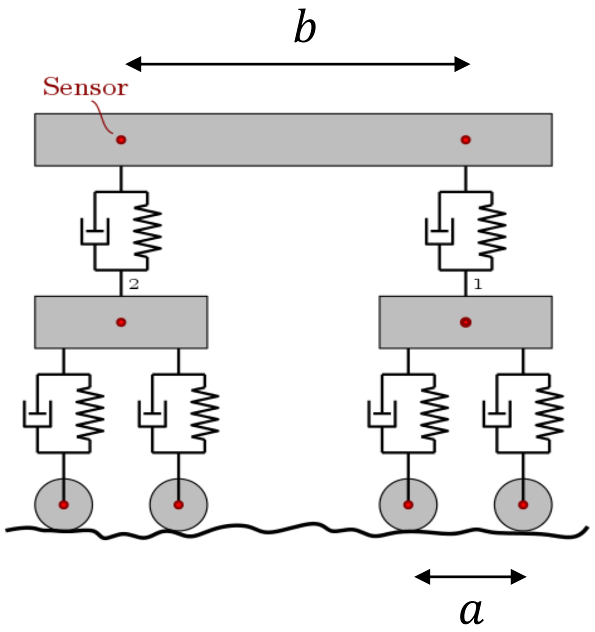
$$\dot{\boldsymbol{y}} = \frac{d\boldsymbol{y}}{dt} = \frac{d}{dt}\begin{pmatrix} x \\ \dot{x} \end{pmatrix} = \begin{pmatrix} \dot{x} \\ \ddot{x} \end{pmatrix} = \begin{pmatrix} \dot{x} \\ -\frac{d}{m} \cdot \dot{x} - \frac{c}{m} \cdot x + \frac{d}{m} \cdot \dot{u} + \frac{c}{m} u \end{pmatrix} = F(\boldsymbol{y}, t)$$

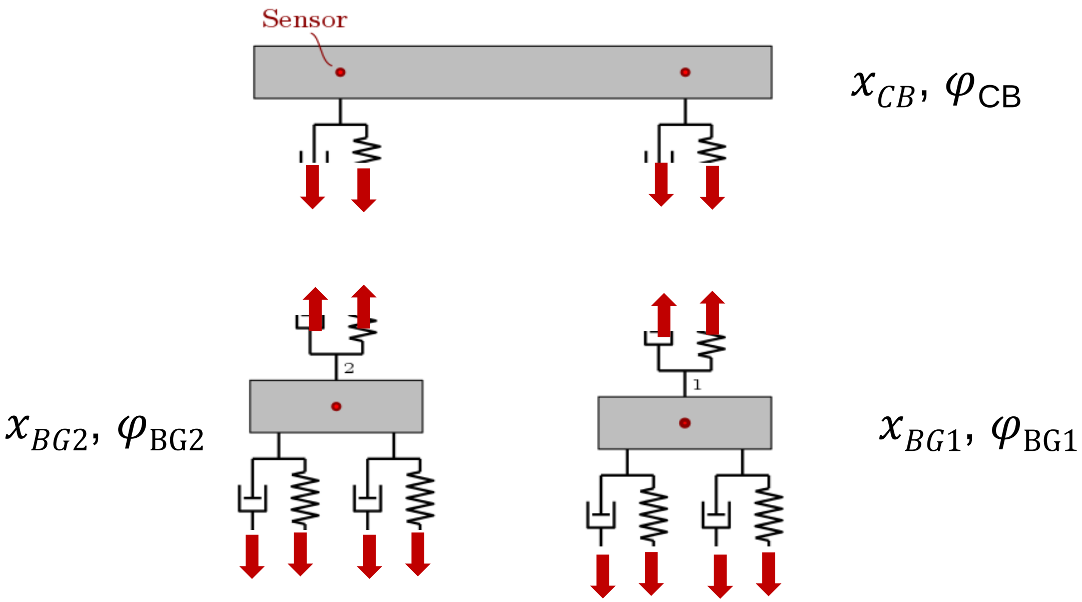
Can be solved with standard solvers in matlab/python

SIEMENS

More complex models

6 Degrees of freedom



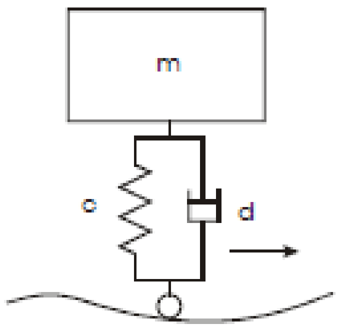


$$\boldsymbol{y} = (x_{CB}\ \varphi_{\mathrm{CB}}\ x_{BG1}\ \varphi_{\mathrm{BG1}}\ x_{BG2}\ \varphi_{\mathrm{BG2}}\ \dot{x}_{CB}\ \dot{\varphi}_{\mathrm{CB}}\ \dot{x}_{BG1}\ \dot{\varphi}_{\mathrm{BG1}}\ \dot{x}_{BG2}\ \dot{\varphi}_{\mathrm{BG2}})$$

$$\dot{\boldsymbol{y}} = \boldsymbol{F}(\boldsymbol{y}, \boldsymbol{t}, \boldsymbol{c}_{11}, \boldsymbol{c}_{12}, \dots, \boldsymbol{d}_{11}, \boldsymbol{d}_{12}, \dots, \boldsymbol{a}, \boldsymbol{b}, \dots)$$

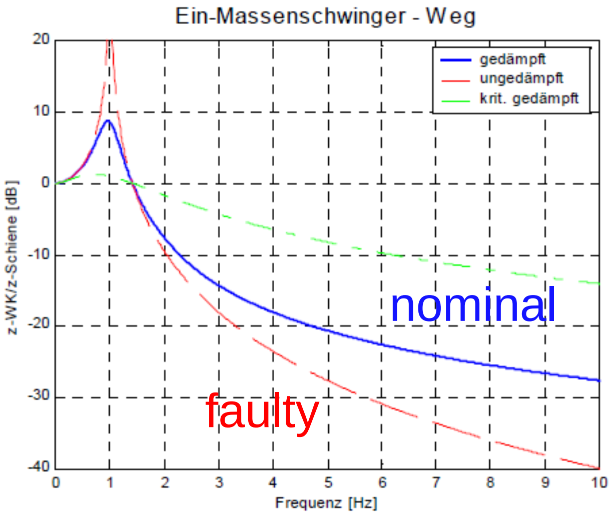
SIEMENS

Transferfunctions of Analogous Model



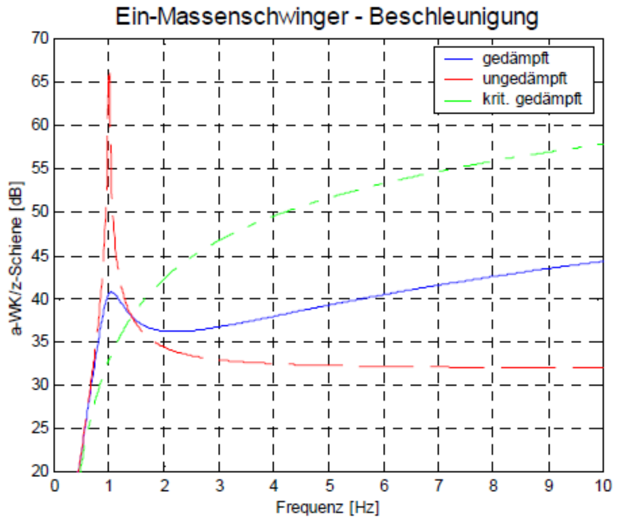
$$\ddot{x} + 2\delta\dot{x} + \omega_0^2 x = r_0 \ \cos(\Omega\ t)$$

One Mass Oscillator Displacement

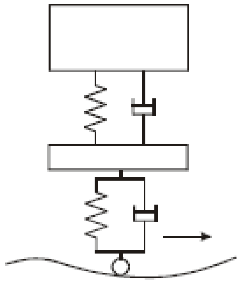


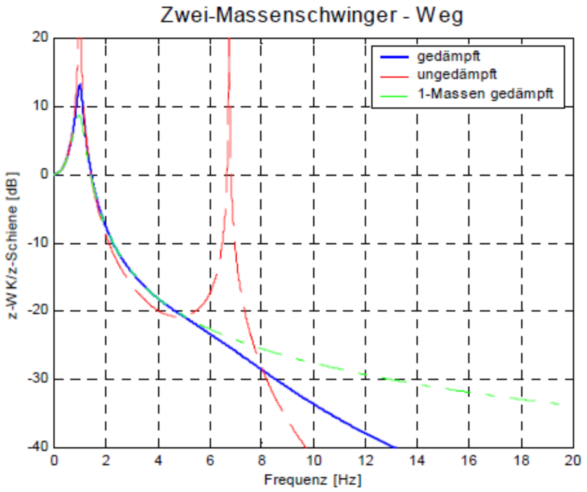
v=280 b=10 a=1.3 f=1.0 DG=0.2 k=0.5

One Mass Oscillator Acceleration



v=280 b=10 a=1.3 f=1.0 DG=0.2 k=0.5

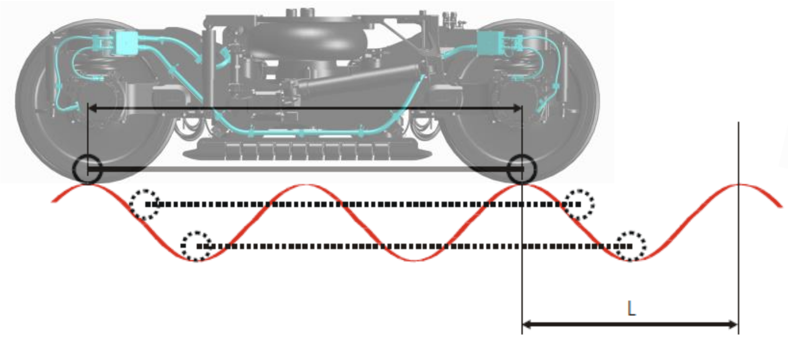


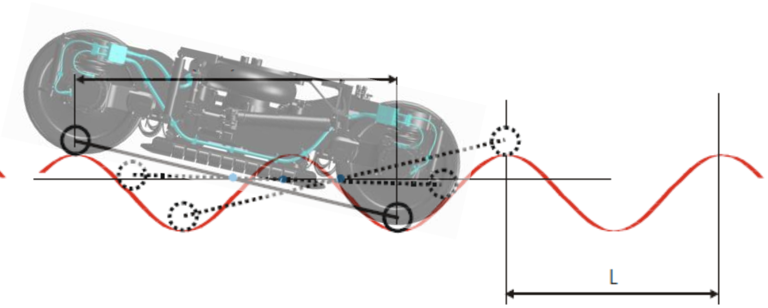


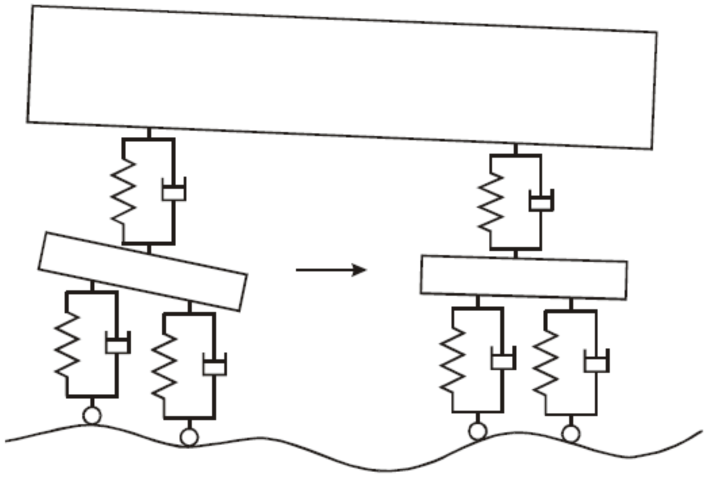
SIEMENS

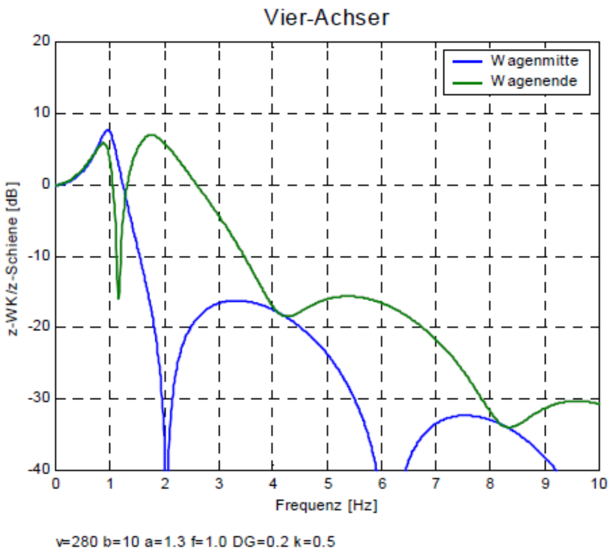
Transferfunction of Bogie Vehicle

Plunge Movement & Pitching Movement









Induces singularities in the transferfunction

SIEMENS